

# **COEN 241 HW1**

## **SYSTEM Virtualization vs OS Virtualization**

Shaunak Galvankar  
W1650127  
sgalvankar@scu.edu

### **1) Configurations:**

**HOST OS- Intel Mac**  
**CPU-2 cores 1.6GHZ**  
**Memory-8gb**

**Docker-**  
**Memory -3.84 GB**

**QEMU-**  
**Memory – 2GB**

### **QEMU:**

**Memory Size : ~2GB**

### **Main Steps to enable a QEMU VM:**

1) Brew install QEMU

2) `sudo qemu-img create ubuntu.img 10G -f qcow2`

3) `sudo qemu-system-x86_64 -hda ubuntu.img -boot d -cdrom ./Downloads/ubuntu-20.04.5-live-server-amd64.iso -m 2046 -boot strict=on`

4) Set up the Ubuntu server following the instructions and Reboot

5) Reboot using `sudo qemu-system-x86_64 -hda ubuntu.img -boot d -m 2046 -boot strict=on`

6) `sudo apt install sysbench`

## **DOCKER:**

Commands used to create my image:

On the Docker container

1) `docker run --rm -it --entrypoint /bin/sh ubuntu:20.04`

- 2) apt update
- 3) apt install sysbench

On my IntelMac Terminal:

- 1) docker commit dc465c4c0009 my\_image\_with\_sysbench

The final image ID for the above image was 7d48b764acb4

```
shaunakgalvankar@Shaunaks-MacBook-Air ~ % docker history my_image_with_sysbench ]
IMAGE          CREATED          CREATED BY
SIZE           COMMENT
7d48b764acb4    15 minutes ago
59.8MB
817578334b4d    8 days ago      /bin/sh -c #(nop)  CMD ["bash"]
0B
<missing>       8 days ago      /bin/sh -c #(nop) ADD file:8faed18d471598732...
72.8MB
shaunakgalvankar@Shaunaks-MacBook-Air ~ %
```

Proof of Experiment

Screenshots of QEMU and Docker working environments

```
QEMU
shaunak@shaunaksqemu:~$ grep MemTotal /proc/meminfo
MemTotal:        2028868 kB
shaunak@shaunaksqemu:~$ sysbench cpu --cpu-max-prime=2000 run
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 2000
Initializing worker threads...

Threads started!

CPU speed:
  events per second: 1583.08

General statistics:
  total time:          10.0089s
  total number of events: 15874

Latency (ms):
  min:                 0.45
  avg:                 0.61
  max:                 65.27
  95th percentile:    0.90
  sum:                 9722.66

Threads fairness:
  events (avg/stddev): 15874.0000/0.00
  execution time (avg/stddev): 9.7227/0.00
shaunak@shaunaksqemu:~$ _
```

```
shaunakgalvankar — com.docker.cli ◀ docker run --rm -it --entrypoint /bin...

Initializing worker threads...

Threads started!

CPU speed:
  events per second: 169.52

General statistics:
  total time:          10.0083s
  total number of events: 1697

Latency (ms):
  min:                 3.20
  avg:                 5.88
  max:                 90.15
  95th percentile:    11.65
  sum:                 9981.08

Threads fairness:
  events (avg/stddev): 1697.0000/0.00
  execution time (avg/stddev): 9.9811/0.00

[# 1]
```

Experiment

# 1)CPU

a)

## CPU Performance max prime=20,000 Docker

```
shaunakgalvankar — com.docker.cli - docker run --rm -it --entrypoint /bin/sh ubuntu:20.04 — 108x40

# sysbench --test=cpu --cpu-max-prime=20000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any
options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 20000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:   278.60

General statistics:
  total time:          10.0057s
  total number of events: 2789

Latency (ms):
  min:                 3.19
  avg:                 3.58
  max:                 47.30
  95th percentile:    4.10
  sum:                 9997.47

Threads fairness:
  events (avg/stddev): 2789.0000/0.00
  execution time (avg/stddev): 9.9975/0.00

#
```

## QEMU CPU MAX Prime =20,000

```
shaunak@shaunaksqemu:~$ sysbench --test=cpu --cpu-max-prime=20000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
  events per second:   81.10

General statistics:
  total time:          10.0323s
  total number of events: 816

Latency (ms):
  min:                 9.98
  avg:                 12.20
  max:                 66.98
  95th percentile:    16.71
  sum:                 9955.98

Threads fairness:
  events (avg/stddev): 816.0000/0.00
  execution time (avg/stddev): 9.9560/0.00

shaunak@shaunaksqemu:~$ _
```

The test ran for 10 seconds for the first iteration on both Docker and QEMU.

QEMU Events/sec =>81.10

Docker Events/sec =>278.6

## b) CPU Performance max prime=60,000 Docker

```
shaunakgalvankar — com.docker.cli • docker run --rm -it --entrypoint /bin/sh ubuntu:20.04 — 108×40
# sysbench --test=cpu --cpu-max-prime=60000
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any
options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

The 'cpu' test requires a command argument. See 'sysbench cpu help'
# sysbench --test=cpu --cpu-max-prime=60000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any
options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 60000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:      59.05

General statistics:
  total time:              10.0059s
  total number of events:   591

Latency (ms):
  min:                     14.88
  avg:                     16.92
  max:                     50.08
  95th percentile:        20.00
  sum:                     10002.18

Threads fairness:
  events (avg/stddev):      591.0000/0.00
  execution time (avg/stddev): 10.0022/0.00
```

## QEMU CPU MAX Prime =60,000

```
shaunak@shaunaksqemu:~$ sysbench --test=cpu --cpu-max-prime=60000 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 60000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:   17.40

General statistics:
  total time:          10.0202s
  total number of events: 175

Latency (ms):
  min:                 46.08
  avg:                 57.01
  max:                 176.48
  95th percentile:    81.48
  sum:                 9976.06

Threads fairness:
  events (avg/stddev): 175.0000/0.00
  execution time (avg/stddev): 9.9761/0.00

shaunak@shaunaksqemu:~$
```

The test ran for 10 seconds for the first iteration on both Docker and QEMU.

QEMU Events/sec =>17.4

Docker Events/sec =>59.05

## c) CPU Performance max prime=60,000 QEMU and Max Time =20s

```
shaunak@shaunaksqemu:~$ sysbench --test=cpu --cpu-max-prime=60000 --max-time=20 run
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 60000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:   20.18

General statistics:
  total time:          20.0286s
  total number of events: 405

Latency (ms):
  min:                 45.57
  avg:                 49.19
  max:                 114.56
  95th percentile:    54.83
  sum:                 19920.98

Threads fairness:
  events (avg/stddev): 405.0000/0.00
  execution time (avg/stddev): 19.9210/0.00

shaunak@shaunaksqemu:~$ _
```



## CPU Performance max prime=60,000 DOCKER and Max Time =20s

```
[# sysbench --test=cpu --cpu-max-prime=60000 --max-time=20 run ]
WARNING: the --test option is deprecated. You can pass a script name or path on the command line without any
options.
WARNING: --max-time is deprecated, use --time instead
sysbench 1.0.18 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Prime numbers limit: 60000

Initializing worker threads...

Threads started!

CPU speed:
  events per second:      59.11

General statistics:
  total time:              20.0089s
  total number of events:  1183

Latency (ms):
  min:                     14.91
  avg:                     16.91
  max:                     58.29
  95th percentile:        20.00
  sum:                     20002.88

Threads fairness:
  events (avg/stddev):      1183.0000/0.00
  execution time (avg/stddev): 20.0029/0.00

# █
```

The test ran for 20 seconds for the first iteration on both Docker and QEMU.

QEMU Events/sec =>20.18

Docker Events/sec =>59.11

## ANALYSIS

1. The VM has lesser events per second processed as compared to the Docker Container for the same set of prime numbers being calculated
2. As we run it for more primes the events/sec rate goes down however docker container still being significantly faster
3. We change the max time for which the tests were running to 20s however the docker container is still 3x faster than the QEMU VM

## 2) FILEIO

```
shaunakgalvankar — com.docker.cli • docker run --rm -it --entrypoint /b

Extra file open flags: (none)
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!


File operations:
  reads/s:                2414.88
  writes/s:               1609.73
  fsyncs/s:              5344.25


Throughput:
  read, MiB/s:            37.73
  written, MiB/s:         25.15


General statistics:
  total time:              10.1050s
  total number of events:  92646


Latency (ms):
  min:                     0.00
  avg:                     1.72
  max:                     70.55
  95th percentile:        5.99
  sum:                    159700.44


Threads fairness:
  events (avg/stddev):    5790.3750/100.39
  execution time (avg/stddev): 9.9813/0.01
```

shaunakgalvankar — com.docker.cli < docker run --rm -it --entrypoint /bin/sh ubuntu:20.04 — 108×40

```
Number of threads: 8
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 24MiB each
3GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...
```

Threads started!

```
File operations:
  reads/s:          2282.39
  writes/s:         1521.27
  fsyncs/s:         4966.65
```

```
Throughput:
  read, MiB/s:      35.66
  written, MiB/s:   23.77
```

```
General statistics:
  total time:                10.0921s
  total number of events:    87517
```

```
Latency (ms):
  min:                      0.00
  avg:                      0.91
  max:                      97.05
  95th percentile:         2.76
  sum:                      79839.90
```

Threads fairness:

File operations:

reads/s:	0.00
writes/s:	365.24
fsyncs/s:	471.35

Throughput:

read, MiB/s:	0.00
written, MiB/s:	5.71

General statistics:

total time:	10.3074s
total number of events:	8505

Latency (ms):

min:	0.02
avg:	1.17
max:	27.13
95th percentile:	3.07
sum:	9971.85

Threads fairness:

events (avg/stddev):	8505.0000/0.00
execution time (avg/stddev):	9.9719/0.00

Initializing worker threads...

Threads started!

Total operations: 13908810 (1390545.21 per second)

13582.82 MiB transferred (1357.95 MiB/sec)

General statistics:

total time:	10.0001s
total number of events:	13908810

Latency (ms):

min:	0.00
avg:	0.00
max:	18.79
95th percentile:	0.00
sum:	4266.24

Threads fairness:

events (avg/stddev):	13908810.0000/0.00
execution time (avg/stddev):	4.2662/0.00

```
shaunakgalvankar — com.docker.cli - docker run --rm -it --entrypoint /bin/sh ubuntu:20.04 — 108x40
Running the test with following options:
Number of threads: 8
Initializing random number generator from current time

Extra file open flags: (none)
128 files, 48MiB each
6GiB total file size
Block size 16KiB
Number of IO requests: 0
Read/Write ratio for combined random IO test: 1.50
Periodic FSYNC enabled, calling fsync() each 100 requests.
Calling fsync() at the end of test, Enabled.
Using synchronous I/O mode
Doing random r/w test
Initializing worker threads...

Threads started!

File operations:
reads/s:          1766.61
writes/s:         1177.24
fsyncs/s:         3867.65

Throughput:
read, MiB/s:      27.60
written, MiB/s:   18.39

General statistics:
total time:       10.0863s
total number of events: 67696

Latency (ms):
min:              0.00
avg:              1.18
max:              33.78
95th percentile: 3.02
sum:              79840.99
```

## ANALYSIS

1. Reducing the number of threads to half has a significant amount of change in the latency but the change in throughput is impacted less comparatively
2. Increasing the write data size has significant amount of change in the throughput however has comparatively lesser change in the latency

## BASH script to automate the sysbench test being run for the experiment

```
Homeworks — vi docker.sh — 108x45

#!/bin/bash
docker run --rm --it --entrypoint /bin/sh ubuntu:20.04
apt update
apt install sysbench
docker commit dc465c4c0009 my_image_with_sysbench
#CPU tests
sysbench --test=cpu --cpu-max-prime=20000 run
sysbench --test=cpu --cpu-max-prime=40000 run
sysbench --test=cpu --cpu-max-prime=60000 run
sysbench --test=cpu --cpu-max-prime=20000 --max-time=20 run
sysbench --test=cpu --cpu-max-prime=20000 --max-time=30 run
sysbench --test=cpu --cpu-max-prime=20000 --max-time=40 run
sysbench --num-threads=16 --test=fileio --file-total-size=3G
#FILEIO tests
sysbench --num-threads=16 --test=fileio --file-total-size=3G
--file-test-mode=rndrw prepare
sysbench --num-threads=16 --test=fileio --file-total-size=3G
--file-test-mode=rndrw run
sysbench --num-threads=16 --test=fileio --file-total-size=3G
--file-test-mode=rndrw cleanup
sysbench --num-threads=16 --test=fileio --file-total-size=6G
--file-test-mode=rndrw prepare
sysbench --num-threads=16 --test=fileio --file-total-size=6G
--file-test-mode=rndrw run
sysbench --num-threads=16 --test=fileio --file-total-size=6G
--file-test-mode=rndrw cleanup
sysbench --num-threads=16 --test=fileio --file-total-size=9G
--file-test-mode=rndrw prepare
sysbench --num-threads=16 --test=fileio --file-total-size=9G
--file-test-mode=rndrw run
sysbench --num-threads=16 --test=fileio --file-total-size=9G
--file-test-mode=rndrw cleanup
sysbench --num-threads=8 --test=fileio --file-total-size=9G
--file-test-mode=rndrw prepare
sysbench --num-threads=8 --test=fileio --file-total-size=9G
--file-test-mode=rndrw run
sysbench --num-threads=8 --test=fileio --file-total-size=9G
--file-test-mode=rndrw cleanup
sysbench --num-threads=4 --test=fileio --file-total-size=9G
--file-test-mode=rndrw prepare
sysbench --num-threads=4 --test=fileio --file-total-size=9G
--file-test-mode=rndrw run
sysbench --num-threads=4 --test=fileio --file-total-size=9G
--file-test-mode=rndrw cleanup
-- INSERT --
```